REMARKS

The claims remaining in the present application are Claims 1-36. Claims 33-36 have been added. No new matter has been added as a result of these amendments.

EXAMINER INTERVIEW SUMMARY

On March 12, 2003, Applicants conducted a telephonic interview with the Examiner. Claims 24, 26, and 27 were discussed with respect to Cmelik et al., U.S. Patent No. 6,031,992. No agreements were reached. Applicants thank to Examiner for conducting this interview.

35 U.S.C. §102

Claims 1 – 32 are rejected under 35 U.S.C. §102(e) as being anticipated by Cmelik et al., U.S. Patent No. 6,031,992 (hereinafter, Cmelik). The rejection is respectfully traversed.

Claim 24

Claim 24 recites:

> A method for scheduling instructions comprising:
> placing a sequence of instructions into one of a plurality of categories based on probability of dependencies; and
> reordering said sequence of instructions based on which of said plurality of categories said sequence of instructions is placed.

Claim 24 recites that a sequence of instructions is placed into one of a plurality of categories based on probability of dependencies. It is respectfully submitted that Claim 24 is not anticipated by Cmelik. Figure 6 and the Specification at page 12, line 5 et seq. provide an example of placing a sequence of instructions into one of

a plurality of categories based on probability of dependencies. However, this is only one example and Claim 24 is not limited to this example.

In rejecting Claim 24, the Examiner cites Cmelik at col. 28, lines 15-25 and 32-35. Applicants respectfully submit that Cmelik fails to disclose placing a sequence of instructions into one of a plurality of categories based on probability of dependencies, as claimed. Rather, Cmelik, in the paragraph containing the cited passages, discloses processing instructions based on the number of times that the instructions are likely to be executed. For example, at line 15 of column 28 Cmelik reads, "steps of reordering and other optimization only occur if it is determined that the particular translation will be run a number of times or otherwise should be optimized." Thus, the reordering takes place based on the frequency with which the particular translation will be run, as opposed to the probability of dependencies, as claimed.

Cmelik at column 28, lines 18-30 describes two ways in which instructions that execute most frequently can be detected, such that the most often run instructions are optimized. One method involves adding host instructions that count the number of times a translation is executed and generate an exception or branch to optimize if the translation is executed a certain number of times (col. 28, lines 18-26). A second method is to interrupt the execution at some frequency or some statistical basis and optimize whatever instruction happens to be executing (col. 28, lines 26-31). Cmelik discloses yet another method to determine which instructions should be optimized, based on the frequency of execution. At col. 28, lines 32-35 Cmelik discloses that certain types of instructions that are likely to be executed

most often, such as those which create loops, are targeted for optimization. Thus, Cmelik discloses several ways to determine which instructions to optimize, based on the likelihood that the translated instructions will be executed. However, Cmelik fails to disclose, in the presently discussed passage or elsewhere, placing a sequence of instructions into one of a plurality of categories based on probability of dependencies, as claimed.

For the foregoing rationale, it is respectfully submitted that Claim 24 is not anticipated by Cmelik. Therefore, allowance of Claim 24 is respectfully submitted.

Claim 26

Claim 26 recites:

> The method of Claim 24, wherein:
> if said sequence of instructions is placed into a category of said plurality of categories in which there are probably no dependencies, said reordering said sequence of instructions comprises reordering without regard to violating a scheduling constraint.

Claim 26 recites that reordering comprises reordering without regard to violating a scheduling constraint if the instructions are placed into a category in which there are probably no dependencies. It is respectfully submitted that Cmelik fails to disclose such a limitation.

The rejection cites Cmelik at col. 32, lines 29-32 in support of rejecting Claim 26. Applicants respectfully submit that this passage does not constitute reordering of instructions, as claimed. Rather, this passage is concerned with updating the official target registers and committing the uncommitted stores in the store buffer at the end of each sequence of host primitive instructions (col. 32, lines 31-38). As the

register updates are happening at the end of a sequence of instructions, it is clear that the sequence of instructions is not being reordered by this process of updating the registers.

For the foregoing rationale, it is respectfully submitted that Claim 26 is not anticipated by Cmelik. Therefore, allowance of Claim 26 is respectfully submitted.

Claims 25-29 and 33-36 depend from Claim 24, which is believed to be allowable for the foregoing reasons. As such, Claims 25-29 and 33-36 are believed to be allowable and their allowance earnestly solicited.

NEW CLAIMS

New Claim 33 recites:

> The method of Claim 24, wherein said plurality of categories comprise at least three separate categories.

Support for Claim 33 can be found in Figure 6. It is respectfully submitted that the limitations of Claim 33 are not disclosed in the cited references.

New Claim 34 recites:

> The method of Claim 24, wherein said plurality of categories comprise at least four separate categories.

Support for Claim 34 can be found in Figure 6. It is respectfully submitted that the limitations of Claim 34 are not disclosed in the cited references.

New Claim 35 recites:

> The method of Claim 24, wherein said plurality of categories comprise categories for:
> no known dependencies;
> known dependencies;
> probably no dependencies; and
> probably dependencies.

Support for Claim 35 can be found in Figure 6. It is respectfully submitted that the limitations of Claim 35 are not disclosed in the cited references.

New Claim 36 recites:

> The method of Claim 24, further comprising:
> determining, for instructions placed in a category in which there probably are dependencies, whether to reorder said instructions or to execute said instructions sequentially.

Support for Claim 36 can be found in Figure 6. Further support can be found in the Specification at page 13, lines 3-10. It is respectfully submitted that the limitations of Claim 36 are not disclosed in the cited references.

CLAIMS 1, 13, and 22

Claim 1 recites, in part:

> reordering a sequence of instructions to run as fast as possible even though the reordered sequence may generate an exception;

Claim 13 recites, in part:

> means for reordering a sequence of instructions to run as fast as possible even though the reordered sequence may generate an exception;

Claim 22 recites, in part:

> reordering a sequence of instructions without concern for a dependency that will result in violation of any scheduling constraints provided that there is no known said dependency;

Claim 1 recites that the sequence of instructions is reordered to run as fast as possible. The instructions are reordered to run as fast as possible by reordering them at will, without concern for dependencies. Referring to Figure 6 of the application, some instructions are reordered at will. As is clear from Figure 6 and the accompanying discussion, dependencies are not taken into consideration when reordering at will. For example, the Specification at page 12, line 18 discloses that instructions that probably have no dependencies are treated as though they in fact have no dependencies and are reordered in a manner to provide the fastest possible execution. Moreover, Figure 6 illustrates cases of reordering at will not only when there are no known dependencies, but also when the probability is between known dependencies and no known dependencies. Thus, even though the reordered sequence may generate an exception, the reordering is done such that the instructions may run as fast as possible.

Cmelik does not disclose reordering a sequence of instructions to run as fast as possible even though the reordered sequence may generate an exception as claimed herein. Cmelik discloses reordering instructions and optimizing (col. 12, lines 3-13). However, Applicants do not understand Cmelik to disclose or suggest reordering to run as fast as possible even though the reordered sequence may generate an exception.

To support the rejection to the language "even though the reordered sequence may generate an exception," the Examiner refers to Cmelik col. 15, lines 1-9, which discloses that there is no provision for dependency checking. However,

this passage is referring only to the hardware portion of the morph host. Referring to Figure 2, this is the enhanced hardware 12. However, even if there is no circuitry in this hardware to detect dependencies, this may be provided for elsewhere (e.g., the software or code morphing 11). Applicants also note that the cited passage also states that the hardware 12 has no circuitry for reordering, optimizing, or rescheduling. However, it is clear from Cmelik that these functions happen somewhere. Thus, it cannot be concluded from this passage of Cmelik whether dependency checking is performed or not. All that is disclosed in the cited passage is that dependency checking is not done in the hardware 12.

For the foregoing rationale, it is respectfully submitted that Claim 1 is not anticipated by Cmelik. As such, allowance of Claim 1 is respectfully submitted. Claim 13 recites similar limitations. As such, allowance of Claim 13 is respectfully submitted.

Claims 2-12, 14-21, 23, and 30-32 depend from Claim 1, 13, and 22 which are respectfully believed to be allowable. As such, allowance of Claims 2-12, 14-21, 23, and 30-32 is earnestly solicited.

CONCLUSION

In light of the above listed amendments and remarks, reconsideration of the rejected Claims is requested. Based on the arguments and amendments presented above, it is respectfully submitted that Claims 1-36 overcome the rejections of record and, therefore, allowance of Claims 1-36 is earnestly solicited.

Should the Examiner have a question regarding the instant response, the Applicants invite the Examiner to contact the Applicants' undersigned representative at the below listed telephone number.

Respectfully submitted,
WAGNER, MURABITO & HAO LLP

Dated: 3/19, 2003

Ronald M. Pomerenke
Registration No. 43,009

Address: WAGNER, MURABITO & HAO LLP
Two North Market Street
Third Floor
San Jose, California 95113

Telephone: (408) 938-9060 Voice
(408) 938-9069 FAX